**MODUL PRAKTIKUM 4**

**REKURENSI DAN PARADIGMA ALGORITMA DIVIDE & CONQUER**



**Disusun Oleh:**

Nama : Kevin Akbar Adhiguna

NPM : 140810170055

Kelas : A

**PROGRAM STUDI S-1 TEKNIK INFORMATIKA**

**DEPARTEMEN ILMU KOMPUTER**

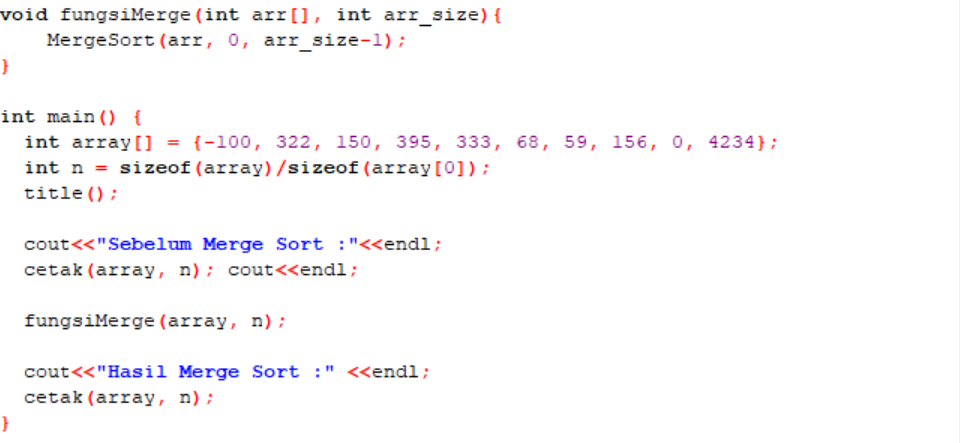
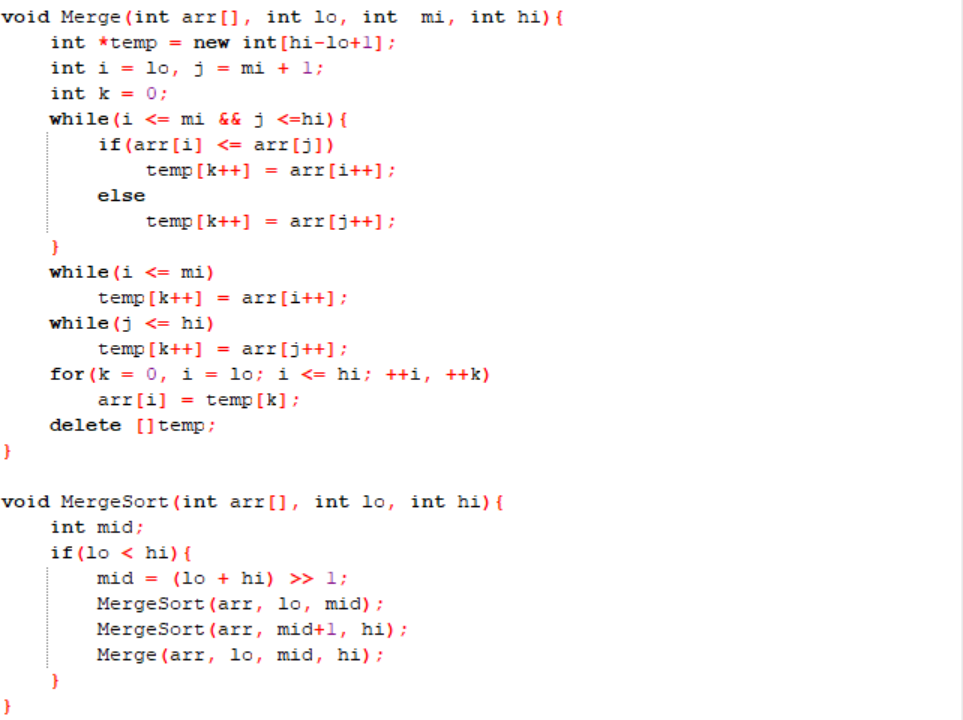
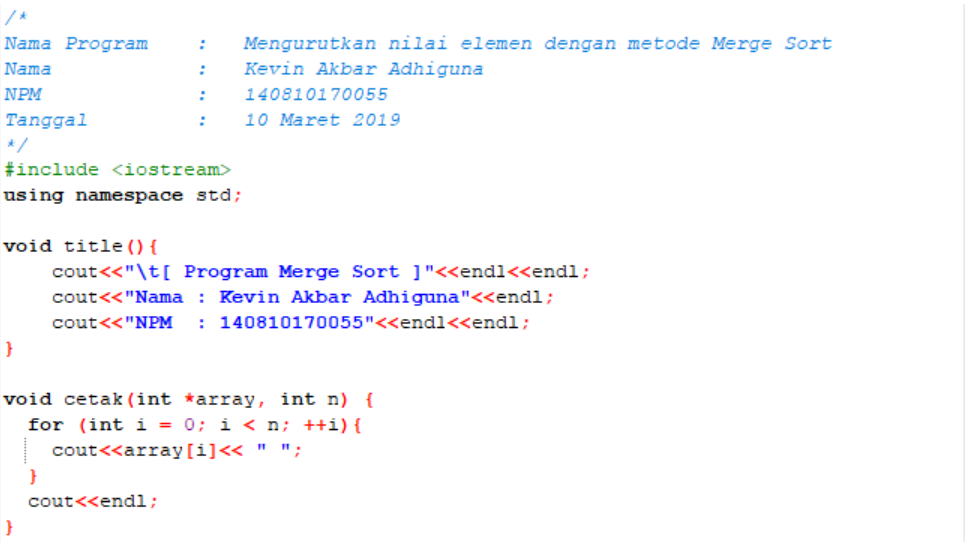
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

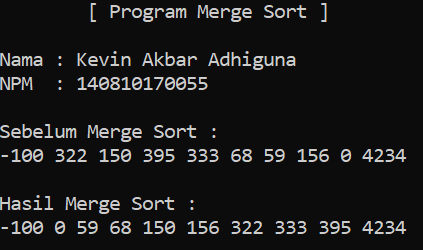
**MARET 2019**

**Studi Kasus 1: MERGE SORT**

Program dalam bahasa C++ :

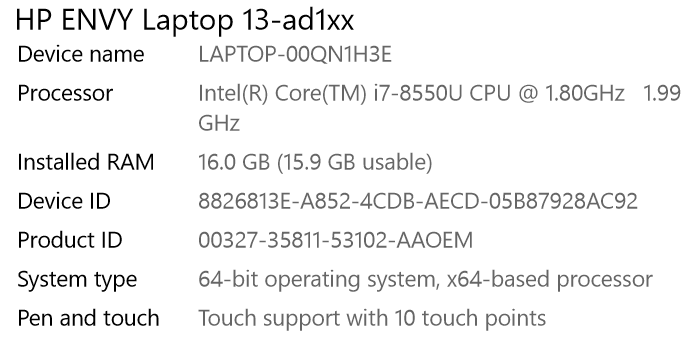


Hasil :



Kompleksitas waktu algoritma merge sort adalah O(n lg n). Cari tahu kecepatan komputer Anda dalam memproses program. Hitung berapa running time yang dibutuhkan apabila input untuk merge sort-nya adalah 20?

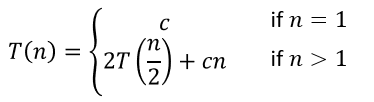
Berikut adalah kecepatan prosesor :



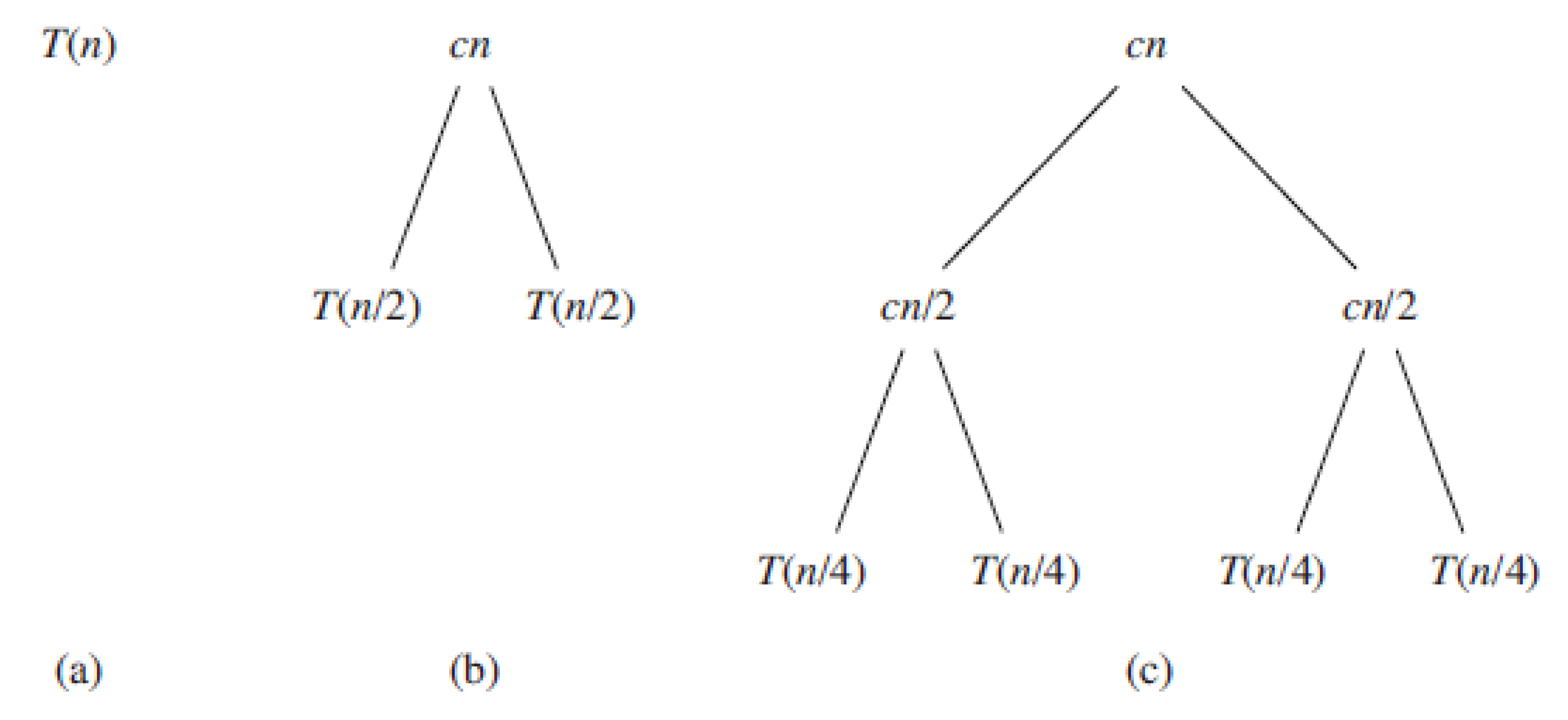
T(26

**Studi Kasus 2: SELECTION SORT**

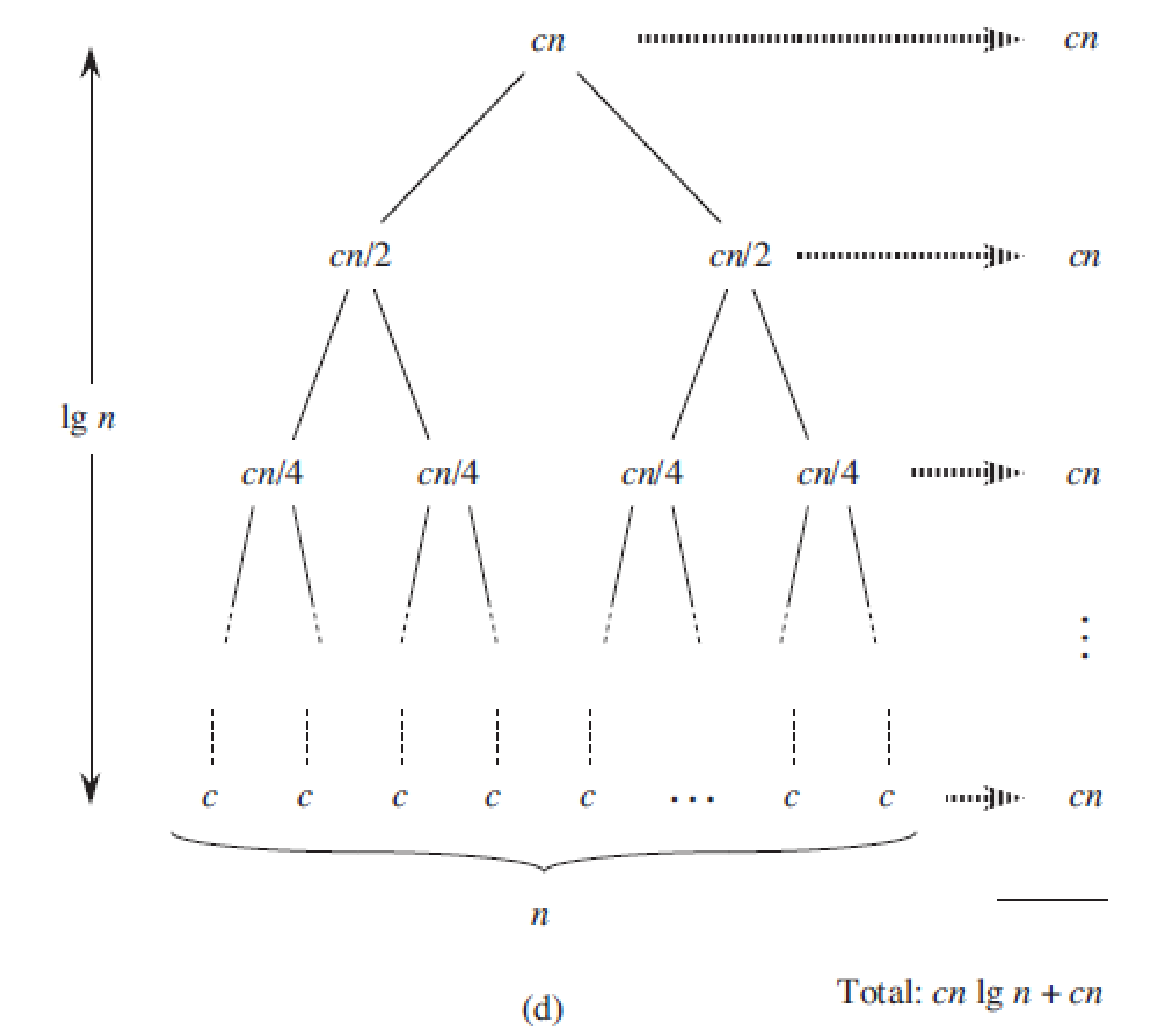
Menentukan T(n) dari rekurensi (pengulangan) selection sort berdasarkan penentuan rekurensi divide & conquer:



Selesaikan persamaan rekurensi T(n) dengan metode **recursion-tree** untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ :

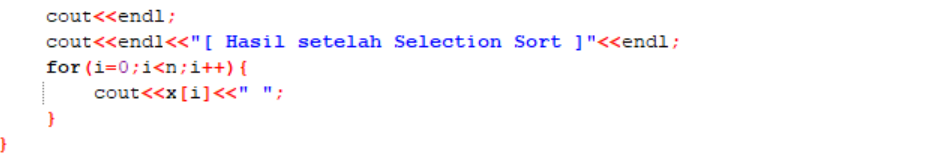
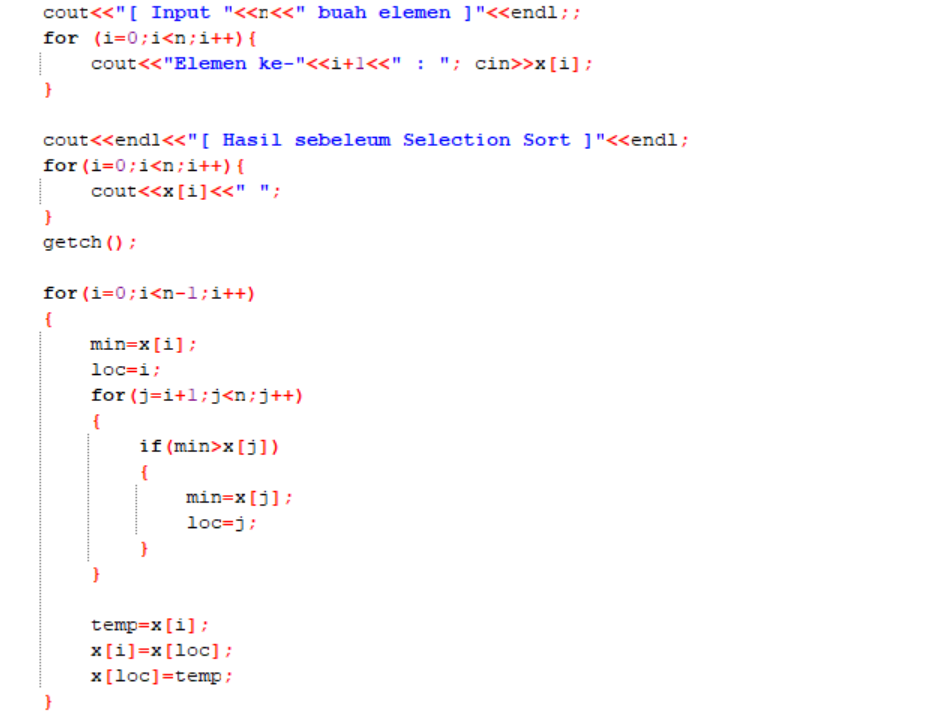
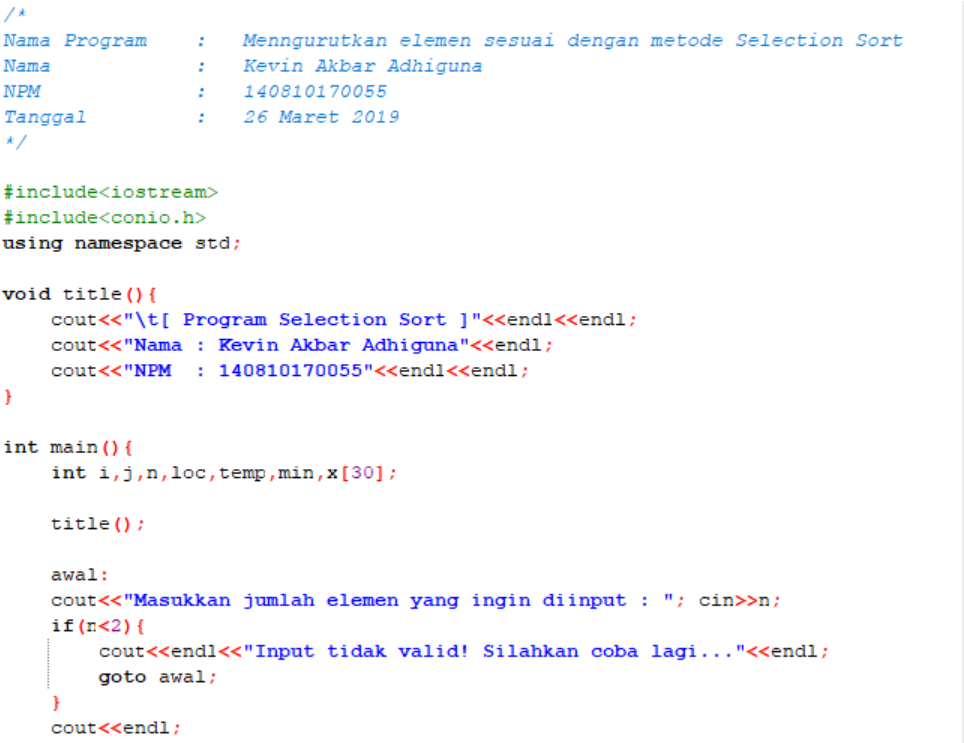
**

(a) menunjukkan , dimana secara progresif dari (b)-(d) untuk membentuk recursion-tree.

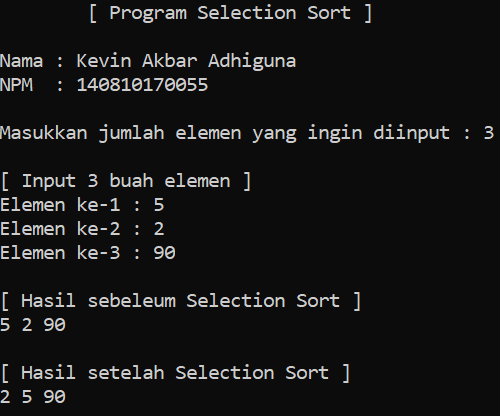


Tree yang diperluas sepenuhnya di bagian (d), mempunya lg n + 1 level (contoh, tree mempunyai tinggi lg n) dan setiap level berkontribusi terhadap total cost dari cn. Oleh karena itu, total costnya menjadi cn lg n + cn, yang mana .

Program dalam bahasa C++ :

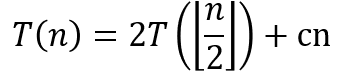


Hasil :



**Studi Kasus 3: INSERTION SORT**

Menentukan T(n) dari rekurensi (pengulangan) insertion sort berdasarkan penentuan rekurensi divide & conquer :



Selesaikan persamaan rekurensi T(n) dengan metode **subtitusi** untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ :

**Langkah – 1:**

untuk :

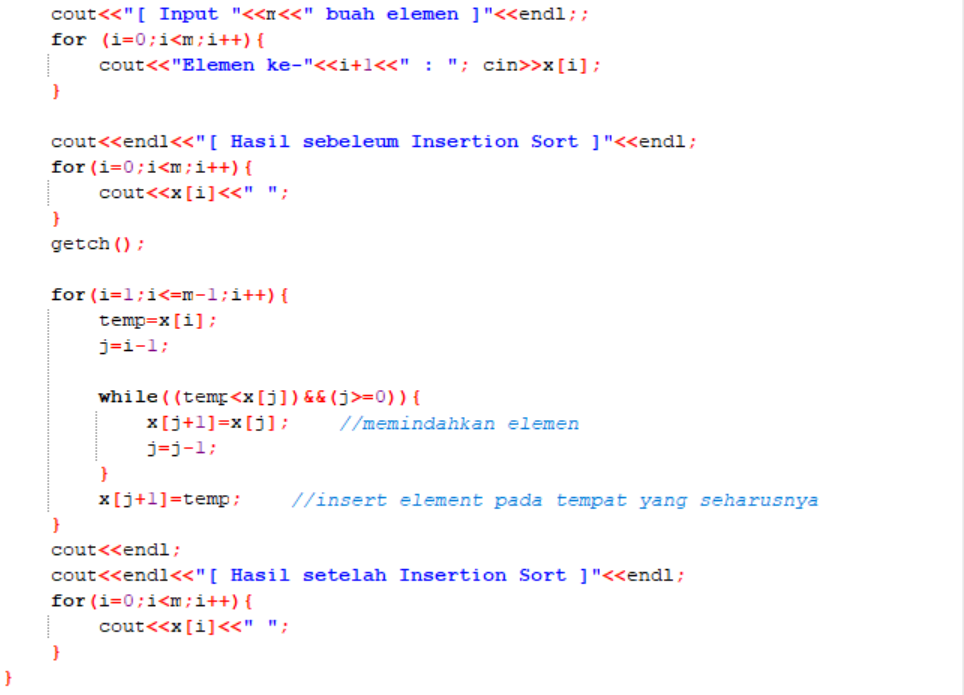
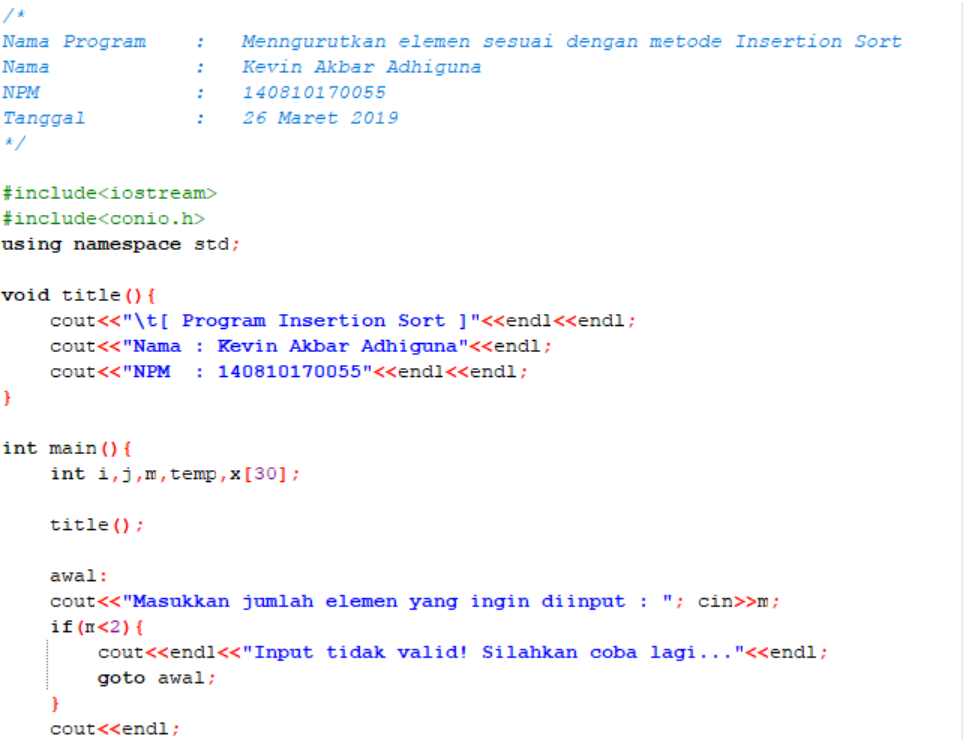
* Solusi tebakannya ->
* Oleh karena itu, harus dibuktikan bahwa untuk , dengan pilihan konstanta positif yang sesuai .
* Perulangan berlaku untuk n = 2, karena dalam kasus ini , dan sehingga kita memiliki .
* Sebagai hipotesis induktif, kita mengasumsikan bahwa ikatan itu berlaku untuk , yaitu T() c lg 🡪 Subtitusikan ini ke dalam rekurensi

**Langkah – 2:**

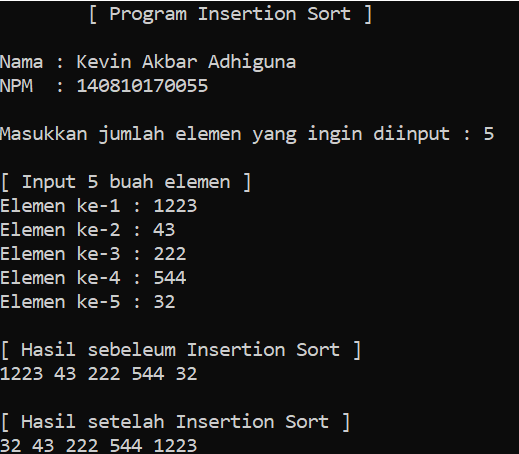
Lalu, menetapkan batas yang diinginkan untuk 𝑇(𝑛), dengan asumsi itu berlaku untuk nilai yang lebih kecil m <n.

Dengan demikian argumen induksi dilengkapi.

Program dalam Bahasa C++ :



Hasil :



**Studi Kasus 4: BUBBLE SORT**

Menentukan T(n) dari rekurensi (pengulangan) insertion sort berdasarkan penentuan rekurensi divide & conquer:

Selesaikan persamaan rekurensi T(n) dengan metode **master** untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ :

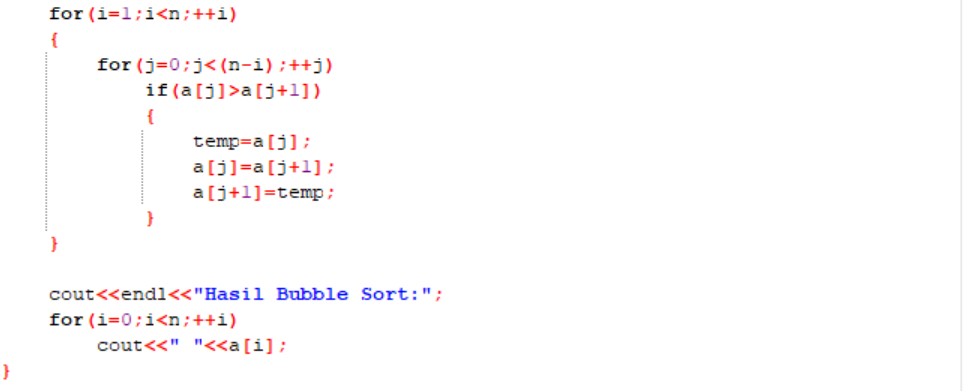
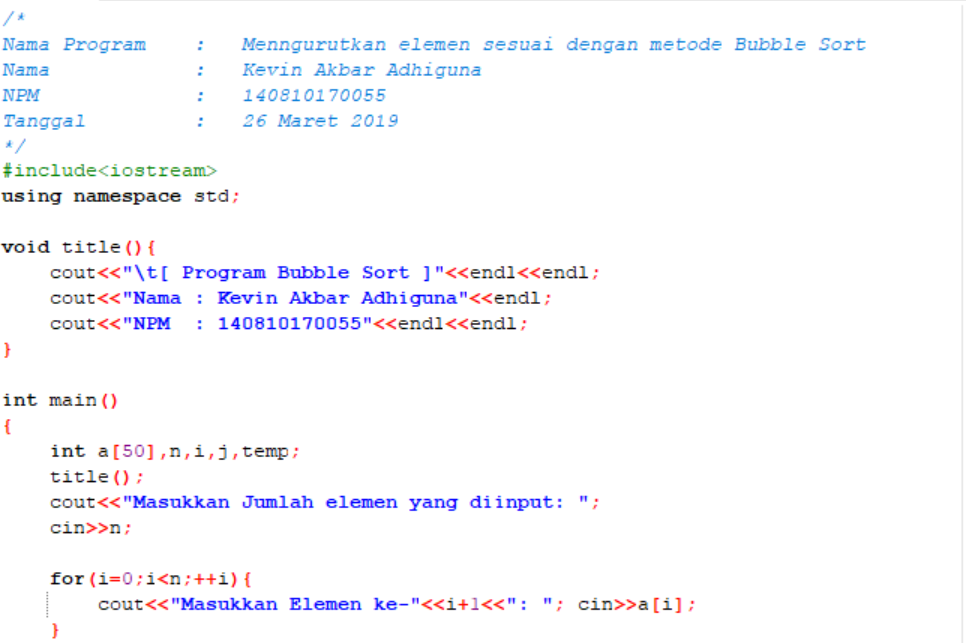
, untuk

Oleh karena itu, berlaku teorema :

“ Jika , untuk konstanta , maka “

Sehingga didapat :

Program dalam bahasa C++



Hasil :

